Corbit

Reference Manual

Kenneth Evans, Jr.

November 1996

Contents

- Overview
- Command-Line Arguments
- Examples
- Orbit Correction
- Singular Value Decomposition
- Response Matrix File
- Configuration File
- Calibration File
- Lattice File
- Output File
- Log File
- Backup File
- Determining Process Variable Names
- Disabling Correctors or Monitors
- Changing the Units or Scaling
- Calculating the Response Matrix from Lattice Data
- Calibrating the Correctors
- Achieving a Desired Orbit
- Compensating for B*Rho
- Previewing the Corrected Orbit
- Predicting the Monitor Readings
- Monitor Averaging
- Monitor Limits
- Monitor Checking
- Monitor Weighting
- Xorbit Simulation Mode

Overview

Corbit is a general-purpose, command-line program to implement orbit correction for the Advanced Photon Source. Its main purpose is to correct the orbit using a file containing the inverse of a response matrix and the names of the monitors and correctors. It will also:

- Calculate the response matrix from lattice data
- Calculate the singular value decomposition of the response matrix
- Remove singular values from the singular value decomposition
- Calculate the inverse from the singular value decomposition
- Use only a subset of the monitors or the correctors
- Preview the corrected orbit
- Match a desired orbit
- Calibrate the response matrix from excitation curve data for the correctors
- Apply an overall calibration to the response matrix
- Compensate for the value of B*Rho
- Weight, average, and check monitors.

Command-Line Arguments

The command line for Corbit is of the form:

```
corbit [options] response-file
```

A response matrix file name is required unless -**g** (generate a response matrix) is specified.

Options:

-**a** *navg* Average over *navg* sets of monitor readings
-**aw** *time* Wait time in sec between averages
-**b** *brho* Use BRho=*brho*; If zero use matrix as is. (Default is to use the current BRho value)
-**bf** *filename* Make backup Burt snapshot of old, new correctors
-**c** *filename* Configuration file is *filename*
-**d** *filename* Desired orbit snapshot file is *filename*
-**e** *nkept* Eliminate all but the *nkept* highest singular values
-**f** *fraction* Only apply *fraction* of calculated correction
-**g** *filename* Generate a response matrix from a lattice file, *filename*
-**H** Make ambiguous names correspond to horizontal
-**h** Help
-**i** Start with inverse matrix from *response-file*
-**j** Match BPM setpoints
-**k** *spec* Multiply matrix by *spec* (=NewUnits/OldUnits)
-**kf** *filename* Get calibrations from a calibration file, *filename*
-**l** *corlimit* Limit to corrector strength is *corlimit*
-**lc** *corchange* Limit to corrector change (Changes will be scaled to be less than *corchange*)
-**lf** *filename* Append statistics to a log file (Create the file if it doesn't exist)
-ms *string* Monitor suffix
-ml *monlimit* Maximum allowed monitor reading
-mc Check monitor validity
-**n** *nsteps* Apply correction in *nsteps* steps
-**o** *filename* Equivalent to -orsio *filename*
-**o**[**r**|**s**|**i**|**o**] *filename*
r Write response matrix to output file, *filename*
s Write SVD matrices to output file, *filename*
i Write inverse matrix to output file, *filename*
o Write output file only, do not correct
(Use *filename*="" or "stdout" for stdout)
-**p** Preview only, do not correct
-**q** Quiet, no output except errors
-**r** Start with response matrix from *response-file*
-**s** Start with SVD U,W,V matrices from *response-file*
-**t** *threshold* Discard singular values < *threshold**maximum
-**V** Make ambiguous names correspond to vertical
-**v** Verbose output
-**w** *waittime* Wait for *waittime* sec between steps (Default is: 5 sec)
-**wf** *filename* Use monitor weights from filename
-**x** Use Xorbit Simulation Mode

The command-line switch names (except -**o** and -**k**) may be longer (-**preview** is the same as -**p**). The predicted orbit is calculated only if -**r** is given.

Examples

Correct the horizontal orbit using the file *par.hrm*:

```
corbit par.hrm
```

Correct the horizontal orbit in five steps using the file *par.hrm*:

```
corbit par.hrm -n 5
```

Correct the horizontal orbit using the file *par.hrm* and specifying the value of B*Rho:

```
corbit par.hrm -b 1.2345
```

Correct the horizontal orbit using the file *par.hrm* (which must contain the response matrix, not just the inverse), keeping the highest 10 singular values, and previewing the expected results without actually correcting:

```
corbit -r -e 10 -p par.hrm
```

Calculate a vertical response matrix file *par.vrm* using lattice data in the file *par.sdds*:

```
corbit -g par.sdds -oro par.vrm
```

Remove flourescent screens from the response matrix file *par.vrm* generated from the lattice file above using the configuration file *par.vconfig*, which only specifies monitors that you want to use for correction, and reuse *par.vrm* for the output file:

```
corbit par.vrm -r -c par.vconfig -oro par.vrm
```

Calibrate the response matrix file *par.vrm* using the calibration file *parcalib.sdds* and write the inverse in *par.vrm* (which can be then be used for correction):

```
corbit par.vrm -r -kf parcalib.sdds -orio par.vrm
```

Perform the above three steps in one command:

```
makeres par.vrm par.sdds par.vconfig parcalib.sdds
```

Orbit Correction

The response matrix is defined as R=dx/dc, where dx are the changes in monitor readings corresponding to changes in corrector settings dc. Orbit correction consists of determining the changes to the corrector settings that will produce a given set of changes in monitor readings (typically those which would make the monitors read zero or alternatively match a saved set of readings). The required corrector settings are those which solve the matrix equation R*dc=dx and are given by dc=R^I*dx, where R^I is the inverse (or pseudo-inverse) of R.

The simplest and most typical use of Corbit is to take an inverse matrix, determine the current monitor readings and corrector settings from the Control System, and then determine the new corrector settings that would make the monitors read zero or match a saved set of values.

Singular Value Decomposition

An M by N matrix R, such as the response matrix, may be written as the product, R=U^T*W*V, of three matrices, where U is an M by N matrix with U*U^T=1, V is an N by N matrix with V^T*V=V*V^T=1, and W is an N by N diagonal matrix with non-negative diagonal elements called the

singular values. The decomposition always exists and is unique except for some permutations of rows and columns. Corbit assumes the singular values along the diagonal of W are ordered in descending order. Sometimes the decomposition is written with U replaced by U^T and V by V^T. The convention used here is that used in Mathematica.[1] Corbit assumes M is greater than or equal to N. The matrix can be extended with rows of zeros to fulfill this requirement if necessary. Note that M > N is typically an over-determined system and M < N is an under-determined system.

The **condition number** of the matrix is the ratio of the largest to the smallest singular value. This value is normally printed by Corbit if the singular value decomposition is calculated.

The **pseudo-inverse** of R is given by $R^I=V^T*W^I*U$, where W^I is a diagonal matrix whose elements are the inverses of the singular values, provided the singular values are non-zero. If some of the singular values are zero and you replace the inverse of these singular values in W^I by zero (even though they are really infinite), it can be shown that the solution, dc, of the equation R*dc=dx using the modified W^I either (1) is the solution with the least norm |dc| if there is a solution or (2) is the best solution in the least-squares sense otherwise; that is, |R*dc-dx| is a minimum.

Replacing the inverses of zero singular values by zero can be viewed as eliminating the rows and columns of the original U, W, and V matrices corresponding to the zero singular values, that is, the rows and columns that would always be multiplied by zero in the product of the three matrices. Even if R is not singular, it is often numerically desirable to eliminate these rows and columns. In this case the product of the modified U^T, W, and V is an approximation to the matrix R, but it may be a good approximation since the terms corresponding to the small singular values did not contribute much to R and may have contributed inordinately to R^I.

You can modify the inverse of the response matrix in Corbit by using the **-e** *nkept* command-line option to eliminate all but the highest *nkept* singular values or the **-t** *threshold* command-line option to remove all singular values less than *threshold* times the largest singular value. In case of a conflict the threshold will determine the number of singular values kept. In practice removing singular values tends to reduce the corrector strength required, but gives a less accurate match to the desired orbit. On the other hand a more accurate match may correspond to overcorrection, where the corrector strengths are large and the orbit is large away from the monitors.

In any event, the singular value decomposition is used in Corbit to calculate the inverse matrix. If the matrix is well-behaved, the resulting inverse should be the same as calculated with any other method. If not, the ability to modify the singular values is useful.

Response Matrix File

A response matrix file is an SDDS[2] file. It may contain pages with five types of matrices:

- Response matrix
- Transpose of the inverse of the response matrix
- U^T, V, and W from the singular value decomposition of the response matrix

---

[1]Mathematica is a registered trademark of Wolfram Research, Inc. and represents a system for doing mathematics by computer.

[2]SSDS stands for Self-Describing Data Sets and is a format developed by Michael Borland to provide a flexible and standard, general purpose, file format. There are routines available for reading and writing SDDS files.

The columns of the file correspond to correctors and the rows correspond to monitors. The singular value decomposition of the response matrix may be written as $R=U^T*Diagonal(W)*V$. The W matrix is a single row with the singular values in order from largest to smallest, and the U^T and V matrices correspond to this ordering of singular values. The U^T, W, and V matrices are either all present or all absent. U^T is written in the file instead of U so that the columns will correspond to correctors throughout the file.

There must be at least one page, and there can be any meaningful combination of the five tables. The first table must be either the Response, the U^T, or the Inverse, so the BPM names may be read from the **BPMNames** column. In order to correct the orbit, only the inverse of the response matrix is required.

There may be other columns and other rows in the pages, provided the means described below is used to define which columns and rows correspond to correctors and monitors.

The corrector names used are defined via the following steps:

1. All the column names are selected except **BPMNames** and **s**.

2. If there is a **CorrectorNames** page in a configuration file, then only the currently selected names that match the selected names in the **CorrectorNames** page are selected (in order of **CorrectorNames**). Otherwise, if there is a **CorrectorNames** parameter (see item 4. below), those names will be used the same way.

3. If there is a **NotCorrectorNames** page in a configuration file, then only the currently selected names that do not match the names in **NotCorrectorNames** are selected. Otherwise, if there is a **NotCorrectorNames** parameter (see item 5. below), those names will be used the same way.

The monitor names used will be defined via the following steps:

1. All the names in the **BPMNames** column will be selected.

2. If there is a **MonitorNames** page in a configuration file, then only the currently selected names that match the selected names in the **MonitorNames** page are selected (in order of **MonitorNames**). Otherwise, if there is a **MonitorNames** parameter (see item 6. below), those names will be used the same way.

3. If there is a **NotMonitorNames** page in a configuration file, then only the currently selected names that do not match the names in **NotMonitorNames** are selected. Otherwise, if there is a **NotMonitorNames** parameter (see item 7. below), those names will be used the same way.

The files will be expected, but not required unless so stated, to contain the following header information:

1. A column of type string for each corrector name. **Required**.

Example:

```
&column name="S1AH1" type=string &end
...
&column name="S40BH1" type=string &end
```

2. A column **BPMNames** of type string for the BPM names. **Required**. This column is required to have the proper BPM names for each of the response, U, and inverse transpose pages that are in the file. The **BPMNames** column is not meaningful for the W and V pages.

Example:

```
&column name="BPMName" type=string &end
```

3. A variable parameter **CorrectionMatrixType** of type string for the matrix type. **Required**. The allowable values are:

    - Response
    - Inverse (Actually the transpose of the inverse)
    - U (From singular value decomposition)
    - W (From singular value decomposition)
    - V (From singular value decomposition)

Example:

```
&parameter name="CorrectionMatrixType" type=string &end
```

4. A parameter **CorrectorNames** of type string indicating what columns are to be used as correctors. This parameter can contain fewer than all the corrector names if the response matrix is the only matrix to be used. Eliminating columns from the other matrices makes them invalid.

Default: Use all columns except **BPMName** and **s**.

Example:

```
&parameter
  name="CorrectorNames" type=string fixed_value="S1AH1 S1AH2 ... S2BH1"
&end
```

5. A parameter **NotCorrectorNames** of type string indicating what additional columns are not to be used as correctors. This parameter can contain some of the corrector names if the response matrix is the only matrix to be used. Eliminating columns from the other matrices makes them invalid.

Default: Use all columns except **BPMName** and **s**.

Example:

```
&parameter
  name="NotCorrectorNames" type=string fixed_value="betax betay"
&end
```

6. A parameter **BPMNames** of type string indicating what rows are to be used as monitors. This parameter can contain fewer than all the monitor names if the response matrix is the only matrix to be used. Eliminating rows from the other matrices makes them invalid.

Default: Use all rows.

Example:

```
&parameter
  name="BPMNames" type=string fixed_value="S1AP1 S1AP2 ... S2BP1"
&end
```

7. A parameter **NotBPMNames** of type string indicating what rows are to not to be used as monitors. This parameter can contain some of monitor names if the response matrix is the only matrix to be used. Eliminating rows from the other matrices makes them invalid.

Default: Use all rows.

Example:

```
&parameter
  name="NotBPMNames" type=string fixed_value="CorrectorStrength S1AH3"
&end
```

8. A parameter **CorrectorLimit** of type double specifying the corrector peak strength.

Default: Use 1 radian (effectively no limit).

Example:

```
&parameter name="CorrectorLimit" type=double fixed_value=.003 &end
```

9. A parameter **SingularValuesKept** of type long specifying the number of singular values kept.

Default: 0

Example:

```
&parameter name="SingularValuesKept" type=long fixed_value=3 &end
```

10. A parameter **CorrectionPlane** of type string specifying the correction plane. The allowable values are `"Horizontal"` or `"Vertical"`. This parameter is not required but is highly desirable.

Default: Correction plane is undefined.

Example:

```
&parameter
  name="CorrectionPlane" type=string fixed_value="Horizontal"
&end
```

11. A parameter **NCorrectors** of type long specifying the number of correctors. This parameter is not required but is highly desirable and is used for error checking.

Default: No error checking will be done.

Example:

```
&parameter name="NCorrectors" type=long fixed_value=10 &end
```

12. A parameter **NMonitors** of type long specifying the number of correctors. This parameter is not required but is highly desirable and is used for error checking.

Default: No error checking will be done.

Example:

```
&parameter name="NMonitors" type=long fixed_value=16 &end
```

13. A parameter **BRhoPVName** of type string specifying the name of the process variable in which the value of BRho is to be found. This parameter is not required, but if it is not present, no compensation for the current value of BRho can be made. It would typically not be included in a calculated response matrix.

Default: ""

Example:

```
&parameter name="BRhoPVName" type=string fixed_value="P:BRhoCalc" &end
```

14. A parameter **ReferenceBRho** of type double specifying the value of BRho for which this response matrix was calculated. This parameter is not required, but if it is not present, no compensation for the current value of BRho can be made. It would typically not be included in a calculated response matrix.

Default: 0.0

Example:

```
&parameter name="ReferenceBRho" type=double fixed_value=1.501038 &end
```

15. A parameter **ResponseMatrixUnits** of type string specifying the units of the response matrix. This parameter is not required, but it is used for checking that an already-converted matrix is not converted again. Currently the only useful value is `"m/A"`, indicating the response matrix is suitable for use in the control system. A calculated response matrix is assumed to have the units `"m/radian"`.

Default: `""`

Example:

```
&parameter
  name="ResponseMatrixUnits" type=string fixed_value="m/A"
&end
```

16. A parameter **MonitorSuffix** of type string specifying a suffix to attach to the **BPMNames** when making process variable names. This parameter is used for the booster and storage ring and has the value ":ms" in those cases. If the string does not contain "x" nor "y", then ":x" or ":y" is appended at the end. Thus, for horizontal ":ms" becomes ":ms:x" and ":ms:x:ErrorCC" remains unchanged. The plane is determined from the corrector names if not specified explicitly. If a monitor suffix has not been supplied, then the monitor name is used as is.

Default: `""`

Example:

```
&parameter
  name="MonitorSuffix" type=string fixed_value=":ms"
&end
```

17. In the **&description** field, the **contents** item contains the machine name and the plane of interest. The **text** field is arbitrary. The correction program makes its own **text** field with a timestamp included. It does not change the **contents** field.

Example:

```
&description
  text="Created by Corbit on Thu Feb 10 13:55:44 1994 from test.hrm"
  contents="PAR Horizontal Response Matrix"
&end
```

Configuration File

A configuration file is an SDDS file used to further configure the response matrix, that is, to supply additional information about which correctors or monitors to use from the response matrix file. There may be up to four pages in the file specifying the names of elements that are correctors, not correctors, monitors, and not monitors. These may be specified by only including a column with the relevant names, or there may be an additional column with flags for each name. The flags can be set to either include or not include the names in the page in which they appear. The configuration file in general and the flags in particular allow the set of correctors and monitors used in the correction to be easily changed. In order to use a configuration file, the response matrix file *must* include the response matrix.

The configuration file may contain other information, but it must include the following parameters and columns:

1. A column **Name** of type string for the element names. **Required**.

Example:

```
&column name="Name" type=string &end
```

2. A column **Flag** of type short for the flags. The value 0 means do not include the names in the page in which they appear, and non-zero means to include them.

Default: All names assumed to be included.

Example:

```
&column name="Type" type=short &end
```

3. A variable parameter **NameType** of type string for the page type. **Required**. The allowable values are:

   - CorrectorNames
   - NotCorrectorNames
   - MonitorNames
   - NotMonitorNames

The meanings should be obvious, and the pages with these parameters have the same effect as the values

for the parameters of the same names in the response matrix file. The name selection logic is described in detail in the response matrix file section. It should be noted that if a CorrectorNames page is present, then only the names selected in that page (and which are also present in the response matrix file) will be used for correction, independently of what other names are specified in the response matrix file. Otherwise, if there is a CorrectorNames parameter in the response matrix file, only those names will be used (if they are present). Otherwise, all the names in the response matrix file will be used. The configuration file takes precedence over the parameters, which in turn take precedence over the names in the file. The NotCorrectorNames page, if it exists, will further limit the names selected by any of the above three means. If it does not exist, then the NotCorrectorNames parameter, if it exists, will further limit the names. Similarly, for monitors.

Example:

```
&parameter name="NameType" type=string &end
```

The following is a sample configuration file which causes the H3 correctors to not be used and the P3P3 monitor to be ignored:

```
SDDS1
&description text="Basic PAR Horizontal Configuration File"
  contents="Correction Program Configuration File" &end
&parameter name=NameType type=string &end
&column name=Name type=string &end
&column name=Flag type=short &end
&data mode=ascii no_row_counts=1 additional_header_lines=1 &end

CorrectorNames
P1H3 0
P1H2 1
P1H1 1
P2H1 1
P2H2 1
P2H3 0
P3H2 1
P3H1 1
P4H1 1
P4H2 1

MonitorNames
P1P4 1
P1P3 1
P1P2 1
P1P1 1
P2P1 1
P2P2 1
P2P3 1
P2P4 1
P3P4 1
P3P3 0
P3P2 1
P3P1 1
P4P1 1
P4P2 1
P4P3 1
P4P4 1
```

Calibration File

A calibration file is an SDDS file used to calibrate a response matrix, that is, to convert a calculated response matrix from corrector readings in m/radian to readings in m/Amperes according to the individual characteristics of the correctors. Such files are derived from magnetic measurements and should be available for each ring or line. A calibration file also contains information on the nominal value of B*Rho and the name of the process variable in which the current value of B*Rho can be found.

The conversion is through the formula:

$$K_nL = C*B_nL/(B*Rho) = (C*B_nL/I)/(B*Rho)*I = R_nL/(B*Rho)*I,$$

where the multipole order n is 0 for corrector dipoles; K_nL is the corrector strength (in radians for dipoles); B_n is the n*th* derivative of the magnetic field; B*Rho is the magnetic rigidity; C is a factor that is close to unity which accounts for individual variations of a corrector from the norm; and I is the current in Amperes. The conversion factor between K_nL (radians) and I (Amperes), apart from the B*Rho factor is R_nL. (R stands for ratio.)

The calibration file may, and probably does, contain other information, but it must include the following parameters and columns:

1. A column **ElementName** of type string for the element names. **Required**.

Example:

```
&column name="ElementName" type=string &end
```

2. A column **RnL** of type double for the R_nL values. **Required**.

Example:

```
&column name="RnL" type=double &end
```

3. A parameter **NominalBRho** of type double specifying the value of B*Rho for which the data are valid. **Required**. This value will overwrite any value of the parameter **ReferenceBRho** found in the response matrix file.

Example:

```
&parameter name="NominalBRho" type=double fixed_value=1.501038 &end
```

4. A parameter **BRhoPVName** of type string specifying the name of the process variable where the current value of B*Rho is to be found. **Required**.

Example:

```
&parameter name="BRhoPVName" type=string fixed_value=PAR:BRhoCALC &end
```

Lattice File

A lattice file is an SDDS file used to calculate or generate a response matrix. Such a file can be created in Xorbit via the File menu using the SDDS File menu choice. This file may contain other information, but it must include the following parameters and columns, depending on which plane is

being calculated:

1. A parameter **nux** or **nuy** of type double for the tune. **Required**.

Example:

```
&parameter name="nux" type=double fixed_value=2.1697 &end
```

2. A parameter **LatticeType** of type string for the lattice type. Allowable values are `"Ring"` or `"Line"`.

Default: `"Ring"`

Example:

```
&parameter name="LatticeType" type=string fixed_value="Ring" &end
```

2. A column **ElementName** of type string containing (possibly along with other element names) the corrector names and the monitor names. **Required**.

3. A column **ElementType** of type string giving the MAD[3] type of the element. For correctors the value is HKIC* or VKIC*, where * indicates any other characters. For monitors it is MONI*. Only the elements matching these types are used. All of the correctors in the file which match the plane being calculated are used, and all the monitors in the file are used. **Required**.

4. A column **betax** or **betay** with the beta values for the element. **Required**.

5. A column **psix** or **psiy** with the phases in radians for the element. **Required**.

Output File

The output file is a response matrix file and has the same specifications as a response matrix file. It may be used as an input file for later invocations of Corbit. The output file may have the same name as the input file, in which case the input file will be overwritten. The output file will contain the matrices after any modifications to them have been made. The matrices included depend on the **-o** switch. See command-line options. If the output filename is specified as "stdout" or "", then the output file will be written to stdout (the terminal) or may be redirected or piped to another program in the usual Unix fashion.

Log File

A log file can be produced with the **-lf** command-line option. The log file will be in SDDS format and will contain the following columns:

Time	Time in sec after 0:00 Jan. 1, 1970, GMT (double)
TimeStamp	Time stamp (string)
PlaneCorrected	Correction plane (string)
CorrSDev	Corrector standard deviation (double)
CorrRMS	Corrector RMS value (double)
CorrAvg	Corrector average value (double)

---

[3]MAD stands for Methodical Accelerator Design and is a widely used code developed at CERN for charged-particle optics in alternating-gradient accelerators and beamlines.

CorrMax	Corrector maximum absolute value (double)
NameOfCorrMax	Corrector with maximum absolute value (string)
MonSDev	Monitor standard deviation (double)
MonRMS	Monitor RMS value (double)
MonAvg	Monitor average value (double)
MonMax	Monitor maximum absolute value (double)
NameOfMonMax	Monitor with maximum absolute value (string)

If the file does not exist, it will be created. Otherwise, the data will be appended.

Backup File

A backup file may be made with the **-bf** command-line option. The backup file will be a valid BURT snapshot file for the correctors. It can be used to restore the correctors.

Determining Process Variable Names

Process variable names, the names which are needed by the control system, are typically more complicated than the common names for monitors and correctors. For example, at the Advanced Photon Source, the corrector P1H1 is set by a process variable, P1H1:CurrentAO, and the monitor, B1C0P1, is read via the process variables, B1C0P1:ms.x and B1C0P1:ms.y. Since it is more convenient to specify the common names and since these are the names generated by computer codes, Corbit has a set of rules for making process variable names from the common names. These rules are:

1. The orbit plane may be specified explicitly by the command-line options **-H** and **-V**. Otherwise, Corbit looks for H or V (but not both) in the corrector names to guess the plane.

2. Corbit keeps track of different process variable names for reading and setting correctors. The suffix ":CurrentAO" is attached to the corrector names for setting, and the same suffix ":CurrentAO" also is attached to the corrector names for reading, even though readbacks really correspond to ":CurrentAI". The reason is that the readbacks are not considered to be sufficiently reliable at present, and the algorithms are more stable in any event if they only deal with the setpoints.

3. If a monitor suffix has been supplied, it is appended to the monitor name. The monitor suffix is used for the booster and storage ring and has the value ":ms" in those cases. If the monitor suffix does not contain "x" nor "y", then ":x" or ":y" is appended at the end, depending on the plane determined by the corrector names. Thus, for horizontal ":ms" becomes ":ms:x" and ":ms:x:ErrorCC" remains unchanged. If a monitor suffix has not been supplied, then the monitor name is used as is. If a monitor suffix has not been supplied, then the monitor name is used as is.

4. If the command-line option **-x** has been specified, then "Xorbit:" is attached to the beginning of the name.

These rules are undoubtedly site-specific. To change them, however, Corbit must be modified.

Disabling Correctors or Monitors

The correctors or monitors may be disabled in one of two ways. The first way is to specify the optional parameters **CorrectorNames**, **NotCorrectorNames**, **MonitorNames**, or **NotMonitorNames** in the response matrix file. The value for each of these parameters is a space-delimited string of the appropriate names.

The second way is to use a configuration file. In the configuration file the appropriate names are

included in the **Name** column of a page corresponding to the variable parameter **NameType**, which may have one of the values `"CorrectorNames"`, `"NotCorrectorNames"`, `"MonitorNames"`, or `"NotMonitorNames"`. Optionally the names to be included can be set with a flag in the **Flag** column. See the Configuration File section for the logic used to determine which names are included.

Changing the Units or Scaling

The response matrix may be multiplied by a constant using the command-line option **-k** *const*, where *const* is the value of the constant. This provides a means of changing the units or scaling. This operation will not change the **ResponseMatrixUnits** parameter in the response matrix file. The only time this parameter is changed by Corbit is when the response matrix is calibrated or calculated.

Calculating the Response Matrix from Lattice Data

The response matrix can be calculated (generated) using the **-g** *filename* command-line option with lattice information in the lattice file denoted by *filename*. You will be prompted for the **CorrectionPlane**, which you *must* specify and for the **CorrectorLimit** and **MonitorSuffix**, which you may optionally specify. The units of the calculated response matrix are m/radian and will appear in the parameter **ResponseMatrixUnits**. All of these parameters, if specified, will appear in the response matrix file that is output. See either the Lattice File or the Examples section for further information on what is required to generate a response matrix. In particular the calculation is different depending on whether the lattice is a ring or a beamline.

Calibrating the Correctors

A calculated response matrix, which corresponds to correctors measured in radians, must be converted to correspond to correctors measured in Amperes to be useful for correction. This means incorporating magnet measurement data relating the angular kick given to the orbit to the current in the power supply that produces this kick. This data is found in SDDS files derived from the magnetic measurements.

A calibration file also has information on the nominal value of B*Rho and the name of the process variable in which the current value of B*Rho may be found.

A calculated response matrix may be calibrated in Corbit by using the **-kf** *filename* command-line option, where *filename* is the name of the calibration file. See either the Calibration File or the Examples section for further information on what is required to calibrate a response matrix. See the section on Changing the Units or Scaling for ways to change the overall scaling factor of the response matrix.

Achieving a Desired Orbit

Instead of calculating corrector strengths that make the orbit zero, it is possible to make the orbit fit a desired orbit from a **Burt**[4] snapshot file using the **-d** *filename* command-line option, where *filename* is the name of the **Burt** snapshot file. The file should have a row for the process variable name of each monitor in the response matrix file. (The process variable name is generated by Corbit from the monitor name according to rules specific to APS naming conventions and is not the same as the monitor name.)

It is also possible to match BPM setpoint process variables, if they exist, by using the **-j** command-line option. (The **-j** option is not mnemonic; it was all that was left.) These process variables are assumed to have the same names as the BPM process variables with ":SetpointAO" appended. Their

[4]Burt stands for the APS Backup and Restore Tool.

values should have been set to the desired orbit values, for this option to be useful.

Compensating for B*Rho

The correction can take into account the current value of B*Rho or not as desired. This capability allows using a response matrix calculated or measured at a (necessarily) single value of B*Rho to be used when the machine is at a different value. Especially during comissioning the value of B*Rho is likely to change.

The requirements for using the current value of B*Rho are knowing the process variable name, **BRhoPVName,** where the current value is to be found and knowing the value of B*Rho, **BRhoReference,** for which the response matrix is valid. They are entered into the response matrix file automatically when the response matrix is calibrated, or they may be entered by editing the file by hand. If a calibration file is used, the values of these parameters in the calibration file override any values that may have appeared in the input response matrix file. (The value which will be used for **ReferenceBRho** is obtained from the parameter **NominalBRho** in the calibration file in this case.)

The default for correction is to use the current value of B*Rho as found in the **BRhoPVName** process variable. In this case the calculated corrector strengths are multiplied by the ratio of the reference to the current values of B*Rho to compensate for the change in B*Rho.

If the -**b** *brho* command-line option is specified and *brho* is not zero, then *brho* will be used for the current value instead of getting it from the process variable. If *brho* is zero, then the matrix will be used as it is without any compensation for B*Rho. This is a way of overriding the default. The matrix will also be used without compensation if **BRhoReference** is zero, if **BRhoPVName** is missing from the response matrix file, or if the current value of B*Rho from the process variable is zero.

Previewing the Correction

The correction may be previewed using the -**p** command-line option. Data on the corrector settings will be displayed, but the correction will not actually take place. It is also possible to preview the predicted monitor readings. This requires the -**r** command-line option as well.

Predicting the Monitor Readings

The predicted monitor readings may be displayed in the summary of the correction which is normally printed to the terminal. Prediction of the orbit requires the response matrix. Correction does not require the response matrix, only the inverse matrix. Consequently, the -**r** command-line option must be specified and there must be a response matrix in the response matrix file in order for the monitor readings to be predicted. Prediction is automatic if the response matrix is read, either because of -**r** or by default. It, along with the rest of the summary, may be supressed with the -**q** command-line option.

Monitor Averaging

Monitor averaging may be specified with the **-a** command-line option. In this case the monitor readings will be averaged over the specified number of readings. The time between readings may be specified by the **-aw** option.

Monitor Limit

A monitor limit may be specified with the **-ml** command-line option. An error message will be printed for all monitors whose absolute readings are over this limit, and the correction will be aborted.

Monitor Checking

Monitor checking may be turned on with the **-mc** command-line option. The status process variables for each monitor will be checked and if any are not valid, the correction will be aborted. The suffix for the status process variables is hard-wired into the code, and the logic is almost surely site dependent.

Monitor Weighting

The monitors may be weighted with the **-wf** command-line option. More highly weighted monitors will be corrected more than less highly weighted monitors. The weights are specified in a weight file in SDDS format. There are two required columns, a string column named **BPMName** and a double column named **Weight**. Any monitors not given a weight in the weight file will be assigned a default weight of 1.0. If weighting is used, the response matrix file will have a short parameter named **Weighted** with value 1 (True). There will be an additional double column named **Weight**. These will be written to the outputput file if **-wf** is specified, and if they are in the input file and Weighted is 1, weighting will be used for the correction. The value of **Weighted** should not be changed manually. (If the response matrix was created with weighting, it would not make sense to use it as unweighted.) The weights will only appear on the **Inverse, SVD**, **W**, and **Response** pages of the file. The other pages may not have the right number of rows.

Xorbit Simulation Mode

Xorbit can in principle be made to simulate the orbit for any of the lattices in the APS, although not all of them have been implemented at this time. To do this there is a special database of process variables in an IOC. These process variables closely resemble the real process variables for monitors and power supplies and have the same names with “Xorbit:” attached. When set points for magnet power supplies are changed via the control system, Xorbit is notified and calculates the appropriate orbit. It then supplies the new readings to the read backs for the power supplies and monitors. This process takes about 4 seconds for the over 3000 process variables for the storage ring, but is essentially instantaneous for the PAR.

The simulation includes the effects of random magnet displacements and monitor offsets and includes nonlinear effects from the sextupoles. This simulation capability allows physics application codes to be tested and debugged on something resembling a real system without disturbing the real system or before the real system is operational.

Corbit may be run in Xorbit simulation mode by specifying **-x** on the command line. In this mode it attaches “Xorbit:” to the front of all process variable names. Consequently, correction may be tested with Xorbit with no change to the response matrix files for a lattice that has been implemented in Xorbit (provided the actual names and the names in the Xorbit database follow the same naming conventions).

Xorbit calculates magnet strengths in theoretical units (e.g., radians for corrector magnets) whereas the real process variables correspond to power supply currents (Amperes). The relationship between the current and the strength depends on the magnet hardware and the energy of the beam. The Xorbit database has been designed to convert between theoretical units and Amperes, taking these relationships into account. The relationship between magnet strength and current is taken to be linear in this databaset. This is a good approximation for correctors, but may be inaccurate for the larger magnets away from their reference values.